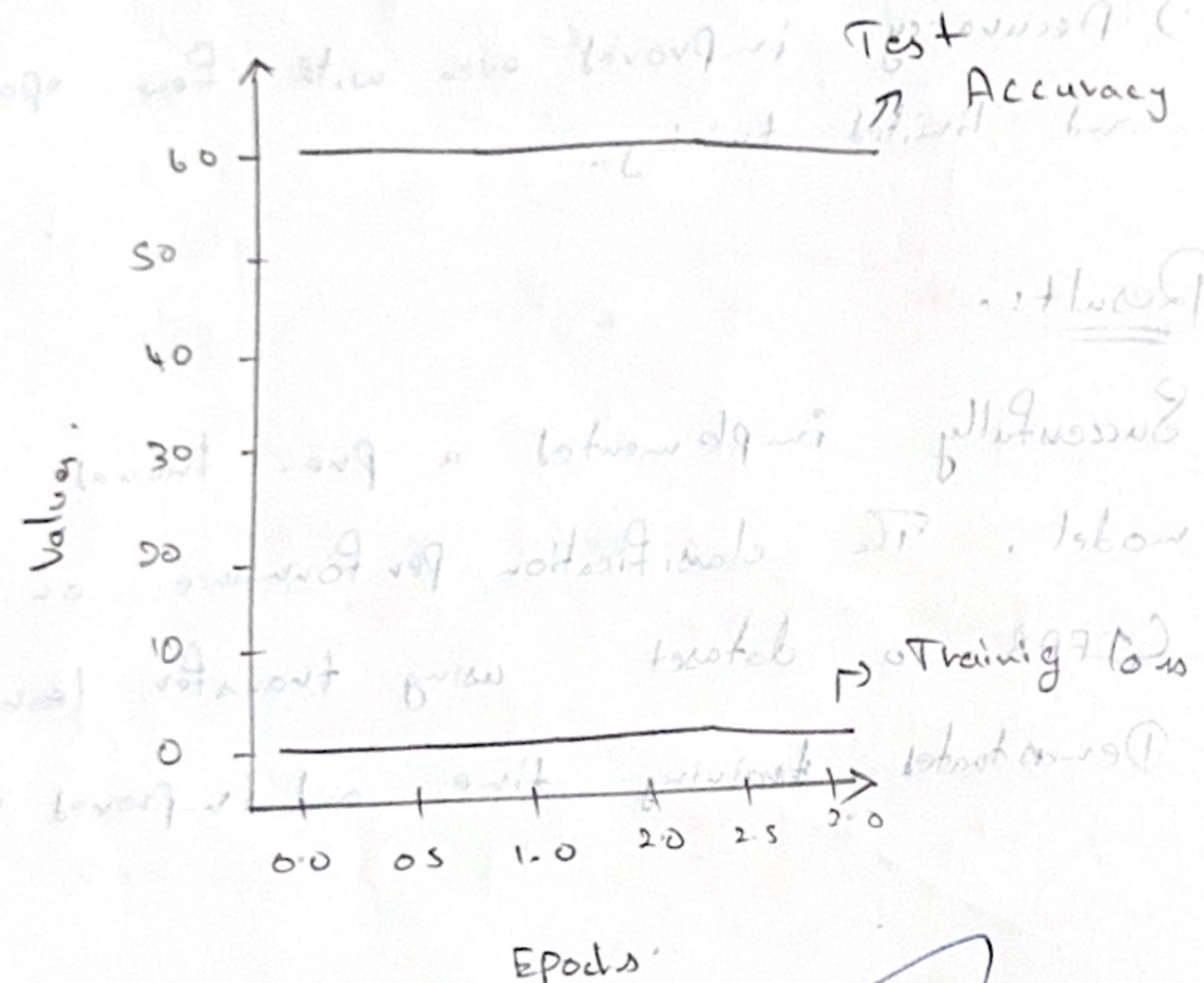


Exp:14:-

Implement a Pre-Defined training CNN model

AIM:-

To implement a pre-trained CNN model (ResNet18) as a feature extractor using for image classification using training learning in pytorch.

objective:-

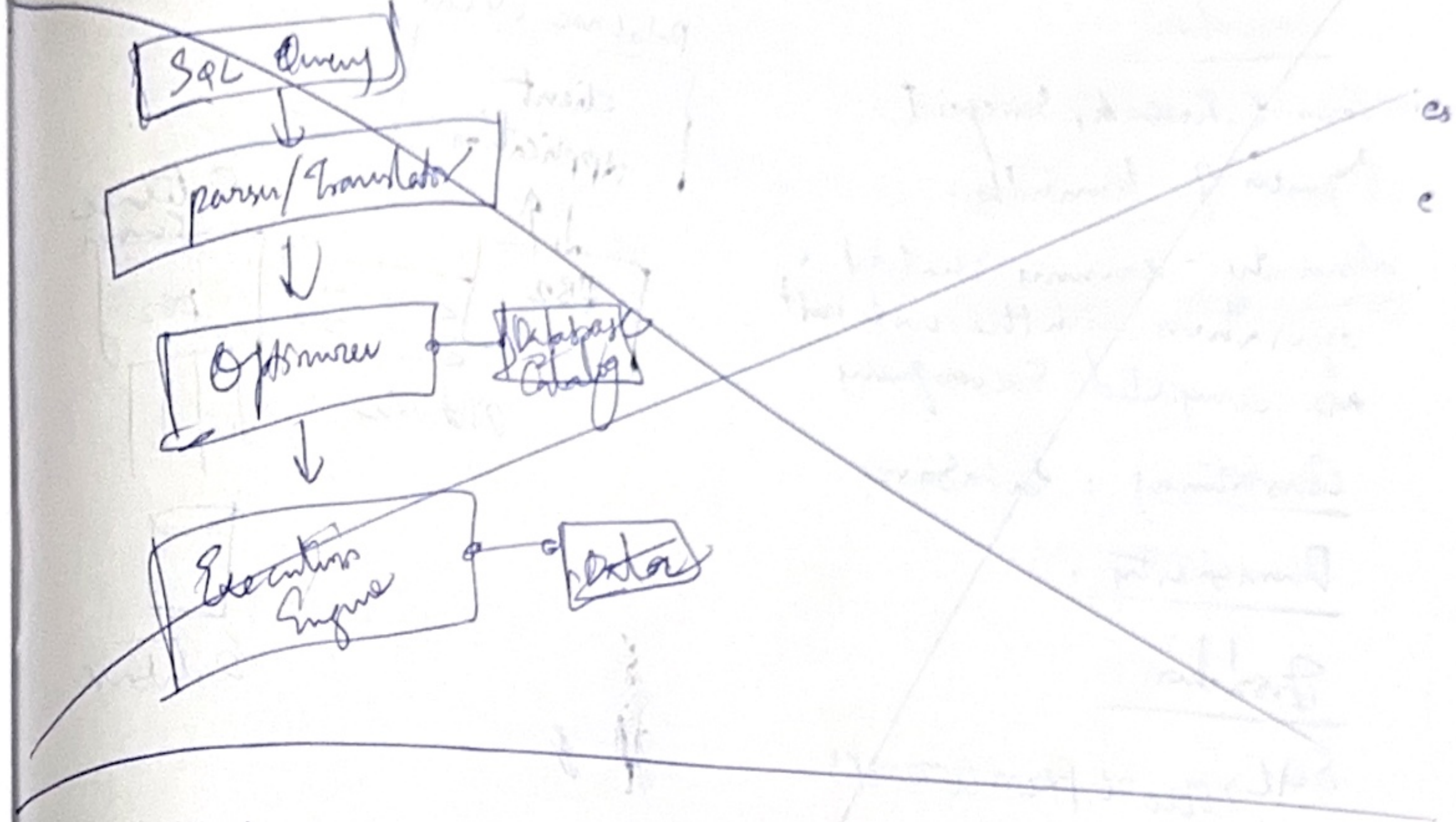
1.) Use a pre-trained CNN model trained on imageNet.
2.) Freeze the convolution layer to use them as feature extractor.
3.) Replace the classifier for our custom dataset.
4.) Train and evaluation the model using classification Report.

Pseudocode:-

```
1.) Import torch, torchvision and others...
2.) Load CIFAR-10 dataset and apply transformation.
3.) Load Pre-trained ResNet-18 model.
4.) Freeze all convolutional layer.
5.) Replace final layer with new classifier for 10 CIFAR.
6.) Train only the classifier layer.
7.) Generater Evaluate the model.
8.) Plot the accuracy point.
9.) Record observation and result.
```

classification Report:-

	Percision	recall	F1-Score	Support.
Airplane.	0.80	0.82	0.81	1000
Automobile.	0.94	0.83.	0.88	1000
bird	0.82.	0.88	0.74.	1000
Cat	0.73	0.66	0.69	1000
deer.	0.71	0.81	0.71	1000
dog	0.78.	0.77	0.77	1000
frog	0.88	0.81	0.83	1000
horse.	0.75	0.88	- 0.81	1000
ship	0.84	0.88	0.86	1000
truck.	0.84	0.91	-0.88	1000
Accuracy.				
Macro avg	0.81	0.80	0.80	10000
weight avg	0.81	0.80	0.80	10000



observation:-

- The Pre-trained CNN extracted rich image features from CIFAR-10 images.
- Only final layer was trained, reducing time
- The model archieved around 85-90% accuracy after just a few epochs.

Result:-

A Pre-trained ResNet 18 CNN was successfully implemented.